**Review Questions**

**Multiple Choice**

1. A file that data is written to is known as a(n)

a. input file

b. output file

c. sequential access file

d. binary file

2. A file that data is read from is known as a(n)

a. input file

b. output file

c. sequential access file

d. binary file

3. Before a file can be used by a program, it must be

a. formatted

b. encrypted

c. closed

d. opened

4. When a program is finished using a file, it should do this.

a. erase the file

b. open the file

c. close the file

d. encrypt the file

5. The contents of this type of file can be viewed in an editor such as Notepad.

a. text file

b. binary file

c. English file

d. human-readable file

6. This type of file contains data that has not been converted to text.

a. text file

b. binary file

c. Unicode file

d. symbolic file

7. When working with this type of file, you access its data from the beginning of the file

to the end of the file.

a. ordered access

b. binary access

c. direct access

d. sequential access

8. When working with this type of file, you can jump directly to any piece of data in the

file without reading the data that comes before it.

a. ordered access

b. binary access

c. direct access

d. sequential access

9. This is a small “holding section” in memory that many systems write data to before

writing the data to a file.

a. buffer

b. variable

c. virtual file

d. temporary file

10. This marks the location of the next item that will be read from a file.

a. input position

b. delimiter

c. pointer

d. read position

11. When a file is opened in this mode, data will be written at the end of the file’s existing

contents.

a. output mode

b. append mode

c. backup mode

d. read-only mode

12. This is a single piece of data within a record.

a. field

b. variable

c. delimiter

d. subrecord

13. When an exception is generated, it is said to have been \_\_\_\_\_\_\_\_\_\_.

a. built

b. raised

c. caught

d. killed

14. This is a section of code that gracefully responds to exceptions.

a. exception generator

b. exception manipulator

c. exception handler

d. exception monitor

15. You write this statement to respond to exceptions.

a. run/handle

b. try/except

c. try/handle

d. attempt/except

**True or False**

1. When working with a sequential access file, you can jump directly to any piece of data

in the file without reading the data that comes before it.

2. When you open a file that file already exists on the disk using the 'w' mode, the contents

of the existing file will be erased.

3. The process of opening a file is only necessary with input files. Output files are automatically

opened when data is written to them.

4. When an input file is opened, its read position is initially set to the first item in the file.

5. When a file that already exists is opened in append mode, the file’s existing contents

are erased.

6. If you do not handle an exception, it is ignored by the Python interpreter and the program

continues to execute.

7. You can have more than one except clause in a try/except statement.

8. The else suite in a try/except statement executes only if a statement in the try suite

raises an exception.

9. The finally suite in a try/except statement executes only if no exceptions are raised

by statements in the try suite.

**Short Answer**

1. Describe the three steps that must be taken when a file is used by a program.

2. Why should a program close a file when it’s finished using it?

3. What is a file’s read position? Where is the read position when a file is first opened for

reading?

4. If an existing file is opened in append mode, what happens to the file’s existing contents?

5. If a file does not exist and a program attempts to open it in append mode, what

happens?

**Algorithm Workbench**

1. Write a program that opens an output file with the filename my\_name.txt, writes your

name to the file, and then closes the file.

2. Write a program that opens the my\_name.txt file that was created by the program in

question 1, reads your name from the file, displays the name on the screen, and then

closes the file.

**292** Chapter 7 Files and Exceptions

3. Write code that does the following: opens an output file with the filename

number\_list.txt, uses a loop to write the numbers 1 through 100 to the file, and then

closes the file.

4. Write code that does the following: opens the number\_list.txt file that was created

by the code you wrote in question 3, reads all of the numbers from the file and displays

them, and then closes the file.

5. Modify the code that you wrote in question 4 so it adds all of the numbers read from

the file and displays their total.

6. Write code that opens an output file with the filename number\_list.txt, but does not

erase the file’s contents if it already exists.

7. A file exists on the disk named students.txt. The file contains several records,

and each record contains two fields: (1) the student’s name, and (2) the student’s

score for the final exam. Write code that deletes the record containing “John Perz”

as the student name.

8. A file exists on the disk named students.txt. The file contains several records, and

each record contains two fields: (1) the student’s name, and (2) the student’s score for

the final exam. Write code that changes Julie Milan’s score to 100.

9. What will the following code display?

try:

x = float('abc123')

print('The conversion is complete.')

except IOError:

print('This code caused an IOError.')

except ValueError:

print('This code caused a ValueError.')

print('The end.')

10. What will the following code display?

try:

x = float('abc123')

print(x)

except IOError:

print('This code caused an IOError.')

except ZeroDivisionError:

print('This code caused a ZeroDivisionError.')

except:

print('An error happened.')

print('The end.')

**Programming Exercises**

**1. File Display**

Assume that a file containing a series of integers is named numbers.txt and exists on the

computer’s disk. Write a program that displays all of the numbers in the file.

**2. File Head Display**

Write a program that asks the user for the name of a file. The program should display only

the first five lines of the file’s contents. If the file contains less than five lines, it should display

the file’s entire contents.

**3. Line Numbers**

Write a program that asks the user for the name of a file. The program should display the

contents of the file with each line preceded with a line number followed by a colon. The

line numbering should start at 1.

**4. Item Counter**

Assume that a file containing a series of names (as strings) is named names.txt and exists

on the computer’s disk. Write a program that displays the number of names that are stored

in the file. *(Hint: Open the file and read every string stored in it. Use a variable to keep a*

*count of the number of items that are read from the file.)*

**5. Sum of Numbers**

Assume that a file containing a series of integers is named numbers.txt and exists on the

computer’s disk. Write a program that reads all of the numbers stored in the file and calculates

their total.

**6. Average of Numbers**

Assume that a file containing a series of integers is named numbers.txt and exists on the

computer’s disk. Write a program that calculates the average of all the numbers stored in

the file.

**7. Random Number File Writer**

Write a program that writes a series of random numbers to a file. Each random number

should be in the range of 1 through 100. The application should let the user specify how

many random numbers the file will hold.

**8. Random Number File Reader**

This exercise assumes you have completed Programming Exercise 7, *Random Number File*

*Writer*. Write another program that reads the random numbers from the file, display the

numbers, and then display the following data:

• The total of the numbers

• The number of random numbers read from the file

**9. Exception Handing**

Modify the program that you wrote for Exercise 6 so it handles the following exceptions:

• It should handle any IOError exceptions that are raised when the file is opened and data

is read from it.

• It should handle any ValueError exceptions that are raised when the items that are read

from the file are converted to a number.